

in the field can be edited, for example, subject field 303. The variable type is set to the value fieldHiddenText to indicate that the text in the field can be edited but is not echoed to the display. The variable type is set to the value fieldButton to indicate that the field is a standard button, for example, button field 308. The variable type is set to the value fieldRadioButton to indicate that the field is a standard radio button, for example, radio button fields 307. The variable type is set to the value fieldCheckBox to indicate that the field is standard check box, for example, check boxes 305. The variable type is set to the value fieldButtonIcon to indicate that the field displays a button in the shape of the specified icon. The variable type is set to the value fieldOptionButton to indicate that the field displays a button with a specified title string. The variable type is set to the value fieldTime to indicate that the field displays the time of day. The variable type is set to the value fieldDate to indicate that the field displays the date. The variable type is set to the value fieldRect to indicate that a rectangle is drawn around the field. The variable type is set to the value fieldPicture to indicate that the field displays the specified picture, for example, picture 301. The variable type is set to the value fieldVariableData to indicate that the field contains data that is not displayed on the screen. The variable type is set to the value fieldUser to indicate that the field is a user-defined field.--

Please replace the paragraph beginning at page 10, line 15 with the following rewritten paragraph:

--The Form Control Procedure (FCP) is a computer subroutine routine that is called directly by the TREV. The FCP is written by the forms designer to implement form customization. The FCP is a block of code stored as the last entry in the form data structure. In a preferred embodiment, the FCP is written in assembly language or another programming

language that is compiled into machine code. Alternatively, the FCP can be written in a scripting language or pseudo-machine language that is interpreted. The use of a scripting language or pseudo-machine language would facilitate platform independent custom forms. In a preferred embodiment, the FCP has full access to the computer resources. Alternatively, the FCP could be restricted as to the resources used. For example, the FCP could be restricted to the operating system calls available to it. The primary key of the FCP id "FFCP" and the secondary key is zero. The following defines the format of the call to the FCP.--

Please replace the paragraph beginning at page 20, line 15 with the following rewritten paragraph:

--Figures 13 and 14 are an example of a flow diagram of the routine to process the events for an FCP that implements a game of tick-tack-toe. Figure 12 is the flow diagram for the main routine in the FCP for this example. The customized form in this example works as follows. There are nine fields in the form. Each field corresponds to a location in a tick-tack-toe grid. The fields contain either no data, an X, or an O. The first player would click the mouse over one of fields. The FCP detects that this field becomes the current field, draws an X in the field, and sets the data in the field to an X. The first player would request that the mail system send the message to the second player. When the message arrives at the second player, the FCP would draw the tick-tack-toe grid and draw an X in the appropriate field. The second player would move the cursor to a field and click the mouse. The FCP would draw an O in that field. The second player would then send the message to the first player. Play would continue until one of the players wins or all the fields contain an X or an O (a tie). When a player wins, the FCP

draws a line through the winning fields and prohibits the placement of any other X's or O's. The FCP also prohibits placing an X or an O in a field that is already occupied.--

Please replace the paragraph beginning at page 21, line 3 with the following rewritten paragraph:

3
b
--Figures 13 and 14 show the flow diagrams for subroutines formNew and fieldPre that are called by the main FCP routine shown in Figure 12. This example of tick-tack-toe using a customizable form could be made more sophisticated, for example, by allowing a player to change his mind before the mail is sent. Figure 13 shows the flow diagram for subroutine formNew. The only function of this routine is to draw the tick-tack-toe grid after the window is created.--

IN THE CLAIMS

Please cancel claims 2-6, 16, 18, 20, 27, 28 and 30-42 without prejudice.

Please amend claims 7-15, 17, 19, 21-26, and 29 as follows:

5
C
7
7. (Amended) A method in a computer system for processing data using a custom form having a field that has been customized by a user of the computer system, the method comprising:

providing a plurality of defined field types that can be associated with custom fields that can be included in the custom form;

providing a plurality of defined behaviors that can be associated with the custom fields that can be included in the custom form;

receiving user input selecting: